


```

      AAAAAA      LL      LL      000000      CCCCCCCC      BBBB88888
      AAAAAA      LL      LL      000000      CCCCCCCC      BBBB88888
AA      AA      LL      LL      00      00      CC      BB      BB
AA      AA      LL      LL      00      00      CC      BB      BB
AA      AA      LL      LL      00      00      CC      BB      BB
AA      AA      LL      LL      00      00      CC      BB      BB
AA      AA      LL      LL      00      00      CC      BBBB88888
AA      AA      LL      LL      00      00      CC      BBBB88888
AAAAAAAAAA      LL      LL      00      00      CC      BB      BB
AAAAAAAAAA      LL      LL      00      00      CC      BB      BB
AA      AA      LL      LL      00      00      CC      BB      BB
AA      AA      LL      LL      00      00      CC      BB      BB
AA      AA      LLLLLLLLLL  LLLLLLLLLL  000000  CCCCCCCC  BBBB88888
AA      AA      LLLLLLLLLL  LLLLLLLLLL  000000  CCCCCCCC  BBBB88888

```

```

LL      IIIIII      SSSSSSSS
LL      IIIIII      SSSSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SSSSSS
LL      II      SSSSSS
LL      II      SS
LL      II      SS
LL      II      SS
LL      II      SS
LLLLLLLLLL  IIIIII      SSSSSSSS
LLLLLLLLLL  IIIIII      SSSSSSSS

```



```

0000 1      .TITLE  ALLOCB - ALLOCATE DYNAMIC MEMORY
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5 :*****
0000 6 :*
0000 7 :*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 :*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 :*  ALL RIGHTS RESERVED.
0000 10 :*
0000 11 :*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 :*  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 :*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 :*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 :*  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 :*  TRANSFERRED.
0000 17 :*
0000 18 :*  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 :*  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 :*  CORPORATION.
0000 21 :*
0000 22 :*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 :*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 :*
0000 25 :*
0000 26 :*****
0000 27
0000 28 :++
0000 29
0000 30 : FACILITY:  F11ACP STRUCTURE LEVEL 1
0000 31
0000 32 : ABSTRACT:
0000 33
0000 34 :     THESE ROUTINES ALLOCATE AND DEALLOCATE SYSTEM NON-PAGED
0000 35 :     DYNAMIC MEMORY FOR FCP CONTROL BLOCKS.
0000 36
0000 37 : ENVIRONMENT:
0000 38
0000 39 :     STAPLET OPERATING SYSTEM, INCLUDING PRIVILEGED SYSTEM SERVICES
0000 40 :     AND INTERNAL EXEC ROUTINES. NOTE THAT THIS ROUTINE MUST BE
0000 41 :     CALLED IN KERNEL MODE.
0000 42
0000 43 :--
0000 44
0000 45 : AUTHOR:  ANDREW C. GOLDSTEIN, CREATION DATE:  14-DEC-1976  16:25
0000 46
0000 47 : MODIFIED BY:
0000 48
0000 49 :     V03-002 CDS0001      Christian D. Saether      9-Apr-1984
0000 50 :             Reduce BYTLM as well as BYTCNT for WCBs.
0000 51
0000 52 :     V03-001 LMP0017      L. Mark Pilant,        29-Mar-1982  11:00
0000 53 :             Put back in the quota checking.
0000 54
0000 55 :     V02-001 LMP0011      L. Mark Pilant,        23-Feb-1982  15:05
0000 56 :             Remove the quota checking inserted in V02-001. This is for
0000 57 :             V3 FT2 only.

```

```

0000 58 :
0000 59 :      V02-001 LMP0005      L. Mark Pilant,      29-Dec-1981 13:05
0000 60 :      Charge the user for any windows created.
0000 61 :
0000 62 : **
0000 63 :
0000 64 :
0000 65 : INCLUDE FILES:
0000 66 :
0000 67 :      INCLUDE "FCPDEF.MAR"
0000 68 :
0000 69 :
0000 70 : EQUATED SYMBOLS:
0000 71 :
0000 72 : ARG LIST OFFSETS
0000 73 :
00000004 0000 74 BYTES = 4      ; BYTE COUNT DESIRED
00000008 0000 75 TYPE = 8      ; BLOCK TYPE BEING ALLOCATED
00000004 0000 76 ADDRESS = 4    ; ADDRESS OF BLOCK BEING DEALLOCATED
0000 77 :
0000 78      $DYNDEF GLOBAL      ; DEFINE STRUCTURE TYPE CODES
0000 79      $IPLDEF            ; DEFINE SYSTEM IPL NAMES
0000 80      $IRPDEF            ; DEFINE I/O PACKET OFFSETS
0000 81      $JIBDEF            ; DEFINE JOB INFORMATION BLOCK FORMAT
0000 82      $PCBDEF            ; DEFINE PROCESS CONTROL BLOCK FORMAT
0000 83      $PRDEF            ; DEFINE PROCESSOR REGISTER NAMES
0000 84      $RSNDEF            ; DEFINE RESOURCE NAMES
0000 85      $WCBDEF            ; DEFINE WINDOW BLOCK FORMAT
0000 86 :      ; USED ONLY FOR TAGS TO THE BLOCK TYPE
0000 87 :      ; AND SIZE FIELDS
0000 88 :
00000000 89      .PSECT $LOCKEDC1$,NOWRT, LONG
0000 90 :
0000 91 : OWN STORAGE:
0000 92 :
0000 93 BLOCK_TYPE:
07 0000 94      .BYTE DYN$C_FCB,-      ; LIST OF SYSTEM TYPE CODES,
0001 95      DYN$C_WCB,-      ; INDEXED BY FCP TYPE CODE
12 0001 96      DYN$C_VCB,-
11 0002 97      DYN$C_RVT,-
0E 0003 98      DYN$C_MVL,-
03 16 0004 99      DYN$C_AQB
0006 100 :
0006 101      .ALIGN 2

```

.....


```

0008 103 :++
0008 104 :
0008 105 : FUNCTIONAL DESCRIPTION:
0008 106 :
0008 107 : THIS ROUTINE ALLOCATES THE REQUESTED BLOCK SIZE FROM SYSTEM
0008 108 : NON-PAGED DYNAMIC MEMORY. THE BLOCK IS CLEARED, AND THE STANDARD
0008 109 : SIZE AND TYPE DATA IS INSERTED.
0008 110 :
0008 111 : CALLING SEQUENCE:
0008 112 : CALL ALLOCATE (ARG1, ARG2)
0008 113 :
0008 114 : INPUT PARAMETERS:
0008 115 : ARG1: NUMBER OF BYTES TO ALLOCATE
0008 116 : ARG2: TYPE OF BLOCK
0008 117 :
0008 118 : IMPLICIT INPUTS:
0008 119 : NONE
0008 120 :
0008 121 : OUTPUT PARAMETERS:
0008 122 : NONE
0008 123 :
0008 124 : IMPLICIT OUTPUTS:
0008 125 : NONE
0008 126 :
0008 127 : ROUTINE VALUE:
0008 128 : ADDRESS OF BLOCK
0008 129 :
0008 130 : SIDE EFFECTS:
0008 131 : BLOCK ALLOCATED
0008 132 :
0008 133 :--
0008 134 :
0008 135 ALLOCATE::
51 04 AC 003C 0008 136 .WORD ^M<R2,R3,R4,R5> ; SAVE THE USUAL REGISTERS
51 04 AC D0 000A 137 10$: MOVL BYTES(AP),R1 ; GET SIZE ARGUMENT
51 0F C0 000E 138 ADDL2 #15,R1 ; ASSUME 16 BYTE GRANULARITY
51 0F CA 0011 139 BICL2 #15,R1
01 08 AC D1 0014 140 SETIPL #IPL$ SYNCH ; RAISE IPL TO SYNCHRONIZE
44 12 001B 141 Cmpl TYPE(AP),#WCB_TYPE ; IS THE BLOCK TO BE A WCB?
00000000'EF D5 001D 142 BNEQ 30$ ; XFER IF NOT, NO FURTHER CHECKS NEEDED
3C 12 0023 143 TSTL CONTEXT_SAVE ; ELSE CHECK FOR SECONDARY CONTEXT
50 00000000'EF D0 0025 144 BNEQ 30$ ; XFER IF IN SECONDARY CONTEXT, NO CHARGE
05 13 002C 145 MOVL CURRENT_WINDOW,R0 ; GET CURRENT WINDOW ADDRESS
OC A0 B5 002E 146 BEQL 15$ ; IF NONE, NO SHARED WINDOW POSSIBLE
2E 13 0031 147 TSTW WCB$$_PID(R0) ; ELSE CHECK FOR SHARED WINDOWS
50 00000000'EF D0 0033 148 BEQL 30$ ; XFER IF SHARED
50 50 OC A0 3C 003A 149 15$: MOVL IO_PACKET,R0 ; ELSE GET ADDRESS OF THE IRP
50 00000000'FF40 D0 003E 150 MOVZWL IRP$$_PID(R0),R0 ; GET PROCESS INDEX
50 50 0080 C0 D0 0046 151 MOVL @SCH$$_GL_PCBVEC[R0],R0 ; GET PCB ADDRESS
52 20 A0 51 C3 004B 152 MOVL PCB$$_JIB(R0),R0 ; GET JIB ADDRESS
07 14 0050 153 SUBL3 R1,JIB$$_BYTCNT(R0),R2 ; CALCULATE NEW BYTE COUNT
50 D4 0052 154 BGTR 20$ ; XFER IF ENOUGH QUOTA
0054 155 CLRL R0 ; ELSE SET AN ERROR
0057 156 SETIPL #0 ; RESTORE IPL
20 A0 52 D0 0059 157 BRB 40$ ; GO FINISH UP
24 A0 51 C2 005D 158 20$: MOVL R2,JIB$$_BYTCNT(R0) ; SET NEW BYTE COUNT
159 SUBL2 R1,JIB$$_BYTLM(R0) ; REDUCE BYTLM ALSO - THIS IS LONG-LIVED ALL

```



```

        7E      DC 0061 160 30$:  MOVPSL  -(SP)           ; SAVE THE PSL FOR WAIT CALL BELOW
00000000'9F   16 0063 161      JSB    @#EXESALONONPAGED ; GET BLOCK FROM EXEC
        21 50   E9 0069 162      BLBC   R0,50$          ; BRANCH ON FAILURE
                                006C 163      SETIPL #0           ; RESTORE IPL
                                006F 164      ; CLEAN PSL OFF STACK AND
        6E      D0 006F 165      MOVL   R1,(SP)        ; SAVE RETURNED BYTE COUNT
        51      DD 0072 166      PUSHL  R2           ; AND ADDRESS
62 51 00 62 00 2C 0074 167      MOVCS  #0,(R2),#0,R1,(R2) ; ZERO OUT THE BLOCK
                                007A 168      ;
        50      D0 007A 169      MOVL   (SP)+,R0       ; GET BLOCK ADDRESS
        08 AO 8E  F7 007D 170      CVTLW  (SP)+,WCB$W_SIZE(R0) ; PUT IN SIZE WORD
        51      D0 0081 171      MOVL   TYPE(AP),R1     ; GET BLOCK TYPE ARG
OA AO 51 08 AC  FF76 CF41 90 0085 172      MOVSB  BLOCK_TYPE[R1],WCB$B_TYPE(R0) ;
        04      04 008C 173 40$:  RET           ; AND RETURN
                                008D 174      ;
                                008D 175      ; WE GET HERE IF MEMORY IS NOT AVAILABLE
                                008D 176      ;
        50      3C 008D 177 50$:  MOVZWL  #RSNS$ NPDYNMEM,R0 ; GET APPROPRIATE RESOURCE CODE
54 00000000'9F D0 0090 178      MOVL   @#SCH$GL_CURPCB,R4 ; AND PROCESS PCB ADDRESS
00000000'9F   16 0097 179      JSB    @#SCH$RWAIT    ; AND WAIT FOR POOL TO APPEAR
        FF6A   31 009D 180      BRW    10$          ; TRY AGAIN
    
```

.....


```

00A0 182 :++
00A0 183 :
00A0 184 : FUNCTIONAL DESCRIPTION:
00A0 185 :
00A0 186 :     THIS ROUTINE DEALLOCATES THE INDICATED BLOCK OF MEMORY BACK
00A0 187 :     TO THE SYSTEM POOL OF NON-PAGED DYNAMIC MEMORY.
00A0 188 :
00A0 189 : CALLING SEQUENCE:
00A0 190 :     CALL     DEALLOCATE (ARG1)
00A0 191 :
00A0 192 : INPUT PARAMETERS:
00A0 193 :     ARG1: ADDRESS OF BLOCK BEING DEALLOCATED
00A0 194 :
00A0 195 : IMPLICIT INPUTS:
00A0 196 :     NONE
00A0 197 :
00A0 198 : OUTPUT PARAMETERS:
00A0 199 :     NONE
00A0 200 :
00A0 201 : IMPLICIT OUTPUTS:
00A0 202 :     NONE
00A0 203 :
00A0 204 : ROUTINE VALUE:
00A0 205 :     NONE
00A0 206 :
00A0 207 : SIDE EFFECTS:
00A0 208 :     BLOCK DEALLOCATED
00A0 209 :
00A0 210 :--
00A0 211 :
00A0 212 DEALLOCATE::
50 04 AC 003C 00A0 213 .WORD    ^M<R2,R3,R4,R5>           : SAVE REGISTERS
51 08 A0 3C 00A2 214     MOVL     ADDRESS(AP),R0           : GET ADDRESS OF BLOCK
12 0A A0 91 00A6 215     MOVZWL   WCB$W_SIZE(R0),R1          : GET BLOCK SIZE
    2D 12 00AA 216     SETIPL   #IPL$_SYNCH           : RAISE IPL TO SYNCHRONIZE
    0C A0 B5 00AD 217     CMPB     WCB$B_TYPE(R0),#DYN$C_WCB : IS THE BLOCK A WCB?
    28 13 00B1 218     BNEQ     10$                : XFER IF NOT, NO FURTHER CHECKS
00000000'EF D5 00B3 219     TSTW    WCB$S_PID(R0)         : ELSE CHECK FOR SHARED WINDOWS
52 00000000'EF D0 00B6 220     BEQL     10$                : XFER IF NOT A SHARED WINDOW
    52 0C A2 3C 00B8 221     TSTL     CONTEXT_SAVE          : ELSE SEE IF IN SECONDARY CONTEXT
52 00000000'FF42 D0 00BE 222     BNEQ     10$                : XFER IF SO, NO CREDIT
    20 A2 51 C0 00C0 223     MOVL     IO_PACKET,R2           : ELSE GET IRP ADDRESS
    24 A2 51 C0 00C7 224     MOVZWL   IRP$S_PID(R2),R2        : GET PROCESS INDEX
00000000'9F 16 00CB 225     MOVL     @SCH$GL_PCBVEC[R2],R2      : GET PCB ADDRESS
    04 00D3 226     MOVL     PCB$S_JIB(R2),R2          : GET JIB ADDRESS
    00E0 227     ADDL2    R1,JIB$S_BYTCNT(R2)         : CREDIT THE USER
    00E6 228     ADDL2    R1,JIB$S_BYTLM(R2)         : CREDIT BYTLM ALSO.
    00E9 229 10$: JSB     @#EXE$DEANONPAGED          : AND DEALLOCATE THRU EXEC
    00EA 230     SETIPL   #0                          : RESTORE IPL
    00EA 231     RET
    00EA 232
    00EA 233
    00EA 234
    00EA 235     .END

```


ALLOCB
Symbol table

- ALLOCATE DYNAMIC MEMORY

D 6

16-SEP-1984 00:41:55 VAX/VMS Macro V04-00
5-SEP-1984 01:05:45 [F11A.SRC]ALLOCB.MAR;1

ADDRESS	= 00000004			DYN\$C_JNLWCB	= 00000024	G
ALLOCATE	= 00000008	RG	02	DYN\$C_JNL_ABL	= 00000001	G
AQB_TYPE	= 00000005			DYN\$C_JNL_ACBM	= 00000004	G
BITMAP_TYPE	= 00000001			DYN\$C_JNL_ADL	= 00000002	G
BLOCK_TYPE	= 00000000	R	02	DYN\$C_JNL_BCB	= 00000003	G
BYTES	= 00000004			DYN\$C_JNL_BUF	= 00000005	G
CONTEXT_SAVE	*****	X	02	DYN\$C_JNL_BXSTS	= 00000013	G
CURRENT_WINDOW	*****	X	02	DYN\$C_JNL_CWQ	= 00000010	G
DEALLOCATE	= 000000A0	RG	02	DYN\$C_JNL_DB	= 00000006	G
DIRECTORY_TYPE	= 00000002			DYN\$C_JNL_DIOREAD	= 00000015	G
DYN\$C_ACB	= 00000002	G		DYN\$C_JNL_JMT	= 00000009	G
DYN\$C_ACL	= 0000003F	G		DYN\$C_JNL_MSG	= 00000012	G
DYN\$C_ADP	= 00000001	G		DYN\$C_JNL_MSGDATA	= 00000014	G
DYN\$C_AQB	= 00000003	G		DYN\$C_JNL_NDL	= 00000008	G
DYN\$C_BOOTCB	= 00000006	G		DYN\$C_JNL_RC	= 00000011	G
DYN\$C_BRDCST	= 0000001A	G		DYN\$C_JNL_RCPC	= 0000000C	G
DYN\$C_BUFIO	= 00000013	G		DYN\$C_JNL_RM	= 0000000A	G
DYN\$C_CDB	= 00000033	G		DYN\$C_JNL_RRP	= 0000000B	G
DYN\$C_CDRP	= 00000039	G		DYN\$C_JNL_RUL	= 0000000D	G
DYN\$C_CD_BBRPG	= 00000002	G		DYN\$C_JNL_SFT	= 00000007	G
DYN\$C_CD_Cddb	= 00000001	G		DYN\$C_JNL_VCL	= 0000000E	G
DYN\$C_CD_SHDW_WRK	= 00000003	G		DYN\$C_JNL_VLE	= 0000000F	G
DYN\$C_CEB	= 00000004	G		DYN\$C_JPB	= 0000001F	G
DYN\$C_CHIP	= 00000048	G		DYN\$C_KFD	= 00000043	G
DYN\$C_CI	= 00000061	G		DYN\$C_KFE	= 00000018	G
DYN\$C_CIA	= 00000045	G		DYN\$C_KFPB	= 00000044	G
DYN\$C_CIDG	= 0000003B	G		DYN\$C_KFRH	= 00000026	G
DYN\$C_CIMSG	= 0000003C	G		DYN\$C_LC_CHREML	= 00000006	G
DYN\$C_CI_BDT	= 00000001	G		DYN\$C_LC_CLS	= 00000005	G
DYN\$C_CI_FQDT	= 00000002	G		DYN\$C_LC_FPEMUL	= 00000007	G
DYN\$C_CLASSDRV	= 00000064	G		DYN\$C_LC_MP	= 00000003	G
DYN\$C_CLU	= 00000065	G		DYN\$C_LC_MSCP	= 00000008	G
DYN\$C_CLU_BTX	= 00000004	G		DYN\$C_LC_SCS	= 00000004	G
DYN\$C_CLU_CLUB	= 00000003	G		DYN\$C_LC_SYSL	= 00000009	G
DYN\$C_CLU_CLUDCB	= 00000005	G		DYN\$C_LKB	= 00000035	G
DYN\$C_CLU_CLUOPT	= 00000006	G		DYN\$C_LKID	= 00000037	G
DYN\$C_CLU_CLUVEC	= 00000002	G		DYN\$C_LNM	= 00000040	G
DYN\$C_CLU_CSB	= 00000001	G		DYN\$C_LOADCODE	= 00000062	G
DYN\$C_CLU_LCKDIR	= 00000007	G		DYN\$C_LOG	= 0000000B	G
DYN\$C_CONF	= 00000007	G		DYN\$C_LPD	= 00000034	G
DYN\$C_CRB	= 00000005	G		DYN\$C_MBX	= 0000002B	G
DYN\$C_CST	= 00000008	G		DYN\$C_MPWMAP	= 00000004	G
DYN\$C_CXB	= 0000001B	G		DYN\$C_MTL	= 00000019	G
DYN\$C_DCCB	= 00000027	G		DYN\$C_MVL	= 00000016	G
DYN\$C_DDB	= 00000006	G		DYN\$C_NDB	= 0000001C	G
DYN\$C_DPT	= 0000001E	G		DYN\$C_NET	= 00000017	G
DYN\$C_ERP	= 0000003A	G		DYN\$C_NON_PAGED	= 00000001	G
DYN\$C_EXTGSD	= 00000028	G		DYN\$C_ORB	= 00000049	G
DYN\$C_FCB	= 00000007	G		DYN\$C_PAGED	= 00000002	G
DYN\$C_FRK	= 00000008	G		DYN\$C_PBH	= 00000020	G
DYN\$C_GSD	= 00000015	G		DYN\$C_PCB	= 0000000C	G
DYN\$C_IDB	= 00000009	G		DYN\$C_PCBVEC	= 00000001	G
DYN\$C_INIT	= 00000063	G		DYN\$C_PDB	= 00000021	G
DYN\$C_IRP	= 0000000A	G		DYN\$C_PFB	= 00000047	G
DYN\$C_IRPE	= 0000002C	G		DYN\$C_PFL	= 00000023	G
DYN\$C_JIB	= 0000002F	G		DYN\$C_PGD	= 00000066	G
DYN\$C_JNL	= 00000067	G		DYN\$C_PGD_F11BC	= 00000001	G

ALLOCB
Symbol table

- ALLOCATE DYNAMIC MEMORY

E 6

16-SEP-1984 00:41:55 VAX/VMS Macro V04-00
5-SEP-1984 01:05:45 [F11A.SRC]ALLOCB.MAR;1

DYN\$C_PHVEC	= 00000002	G		
DYN\$C_PIB	= 00000022	G		
DYN\$C_PMB	= 00000046	G		
DYN\$C_PQB	= 0000000D	G		
DYN\$C_PRCMAP	= 00000005	G		
DYN\$C_PTR	= 00000025	G		
DYN\$C_RBM	= 00000031	G		
DYN\$C_RIGHTSLIST	= 00000042	G		
DYN\$C_RSB	= 00000036	G		
DYN\$C_RSHT	= 00000038	G		
DYN\$C_RVT	= 0000000E	G		
DYN\$C_SCS	= 00000060	G		
DYN\$C_SCS_CDL	= 00000001	G		
DYN\$C_SCS_CDT	= 00000002	G		
DYN\$C_SCS_DIR	= 00000003	G		
DYN\$C_SCS_HQB	= 0000000B	G		
DYN\$C_SCS_PB	= 00000004	G		
DYN\$C_SCS_PDT	= 00000005	G		
DYN\$C_SCS_RDT	= 00000006	G		
DYN\$C_SCS_SB	= 00000007	G		
DYN\$C_SCS_SPNB	= 00000009	G		
DYN\$C_SCS_SPPB	= 00000008	G		
DYN\$C_SCS_UQB	= 0000000A	G		
DYN\$C_SHB	= 0000002A	G		
DYN\$C_SHMCEB	= 0000002E	G		
DYN\$C_SHMGSD	= 00000029	G		
DYN\$C_SHRBUF IO	= 00000080	G		
DYN\$C_SLAVCEB	= 0000002D	G		
DYN\$C_SPECIAL	= 00000080	G		
DYN\$C_SSB	= 0000001D	G		
DYN\$C_SUBTYPE	= 00000060	G		
DYN\$C_SWPMAP	= 00000003	G		
DYN\$C_TQE	= 0000000F	G		
DYN\$C_TWP	= 00000030	G		
DYN\$C_TYPAHD	= 00000014	G		
DYN\$C_UCB	= 00000010	G		
DYN\$C_UNUSED_2	= 00000041	G		
DYN\$C_VCA	= 00000032	G		
DYN\$C_VCB	= 00000011	G		
DYN\$C_WCB	= 00000012	G		
DYN\$C_WQE	= 0000003E	G		
DYN\$C_XWB	= 0000003D	G		
EXE\$A[ONONPAGED	*****	X	02	
EXE\$DEANONPAGED	*****	X	02	
FCB_TYPE	= 00000000			
HEADER_TYPE	= 00000000			
INDEX_TYPE	= 00000003			
IO_PACKET	*****	X	02	
IPC\$ SYNCH	= 00000008			
IRP\$C_PID	= 0000000C			
JIB\$L_BYTCNT	= 00000020			
JIB\$L_BYTLM	= 00000024			
MVL_TYPE	= 00000004			
PCB\$L_JIB	= 00000080			
PR\$ IPL	= 00000012			
RSN\$ NPDYMEM	= 00000003			
RVT_TYPE	= 00000003			

SCH\$GL_CURPCB	*****	X	02
SCH\$GL_PCBVEC	*****	X	02
SCH\$RWAIT	*****	X	02
TYPE	= 00000008		
VCB_TYPE	= 00000002		
WCBSB_TYPE	= 0000000A		
WCBSL_PID	= 0000000C		
WCBSW_SIZE	= 00000008		
WCB_TYPE	= 00000001		

-----+
! Psect synopsis !
-----+

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
\$AB\$\$	00000000 (0.)	01 (1.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE
\$LOCKEDC1\$	000000EA (234.)	02 (2.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG

-----+
! Performance indicators !
-----+

Phase	Page faults	CPU Time	Elapsed Time
Initialization	48	00:00:00.14	00:00:01.10
Command processing	155	00:00:00.93	00:00:05.72
Pass 1	242	00:00:06.73	00:00:16.30
Symbol table sort	0	00:00:00.89	00:00:01.26
Pass 2	63	00:00:01.56	00:00:04.97
Symbol table output	21	00:00:00.17	00:00:00.58
Psect synopsis output	6	00:00:00.05	00:00:00.98
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	538	00:00:10.48	00:00:30.92

The working set limit was 1350 pages.
36359 bytes (72 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 677 non-local and 7 local symbols.
338 source lines were read in Pass 1, producing 19 object records in Pass 2.
20 pages of virtual memory were used to define 19 macros.

-----+
! Macro library statistics !
-----+

Macro library name	Macros defined
_\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	8
-\$255\$DUA28:[SYSLIB]STARLET.MLB;2	4
TOTALS (all libraries)	12

747 GETS were required to define 12 macros.

There were no errors, warnings or information messages.

MACRO/LIS=I.ISS\$:ALLOCB/OBJ=OBJ\$:ALLOCB MSRC\$:FCPPRE/UPDATE=(ENH\$:FCPPRE)+MSRC\$:ALLOCB/UPDATE=(ENH\$:ALLOCB)+EXECMLS/LIB

CHKSUM LIS
ACPCNTR LIS
CHKPRO LIS
FCPOEF B32
DEACCS LIS
BADSEN LIS
CLENUP LIS
CPYNAM LIS
CHKHDR LIS
COMMON LIS
CREHDR LIS
CREWIN LIS
ALDOB LIS
ACCESS LIS
CHKDMO LIS
DELETE LIS
CREATE LIS
CREFCB LIS